Multichannel Music Separation with Deep Neural Networks

Aditya Arie Nugraha, Antoine Liutkus, Emmanuel Vincent

Multichannel Music Separation with Deep Neural Networks

Aditya Arie Nugraha*†‡, Antoine Liutkus*†‡, and Emmanuel Vincent*†‡
* Inria, Villers-lès-Nancy, F-54600, France
† Université de Lorraine, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54506, France
‡ CNRS, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54506, France
E-mails: {aditya.nugraha, antoine.liutkus, emmanuel.vincent}@inria.fr

***Abstract*—This article addresses the problem of multichannel music separation. We propose a framework where the source spectra are estimated using deep neural networks and combined with spatial covariance matrices to encode the source spatial characteristics. The parameters are estimated in an iterative expectation-maximization fashion and used to derive a multichannel Wiener filter. We evaluate the proposed framework for the task of music separation on a large dataset. Experimental results show that the method we describe performs consistently well in separating singing voice and other instruments from realistic musical mixtures.**

I. Introduction

Music separation is a special case of audio source separation, which aims to recover the singing voice and possibly other instrumental sounds from a musical polyphonic mixture. It has many interesting applications, including music editing/remixing, upmixing, music information retrieval, and karaoke [1]–[3].

Recent studies have shown that deep neural networks (DNNs) are able to model complex functions and perform well on various tasks [4]. Many studies have addressed the problem of single-channel source separation with DNNs. The DNNs typically operate on magnitude or log-magnitude spectra in the Mel domain or the short time Fourier transform (STFT) domain. The DNNs can be used either to predict the source spectrograms [5]–[8] whose ratio yields a time-frequency (TF) mask or directly to predict a TF mask [9]–[13]. The estimated source signal is then obtained as the product of the input mixture signal and the estimated TF mask. Only few of the studies consider the problem of music separation [6], [8], while the others focus on speech separation.

As shown in many works mentioned above, the use of DNNs for audio source separation by modeling the spectral information is extremely promising. However, existing literature lacks a framework to exploit DNNs for multichannel audio source separation. Most of the approaches considered single-channel separation, where the input signal is either one of the channels of the original multichannel mixture signal or the result of averaging over channels. Efforts on exploiting multichannel data have been done by extracting multichannel features and using them to derive a single-channel TF mask [7], [10]. Thus, they do not fully exploit the benefits of multichannel processing as achieved by multichannel filters [1], [2].

In this article, we present a DNN-based multichannel source separation framework where the source spectra are estimated using DNNs and used to derive a multichannel filter through an iterative algorithm. This framework is built upon the state-of-the-art iterative expectation-maximization (EM) algorithm [14], which integrates spatial and spectral models in a probabilistic fashion. This approach was successfully used up to some variants in [15]–[18], but never with DNN models for the sources. We also present the use of multiple DNNs, which are intended to improve the spectra over the iterations. Finally, we present the application of the proposed framework to the separation of professionally-produced music recordings, using a specifically re-engineered version of the dataset used in the 2015 Signal Separation Evaluation Challenge (SiSEC)[1] [19].

The systems for music separation presented in this article are similar to the speech enhancement system in [20]. Beside the difference in the considered data and separation task, in this article we use weighted spatial parameter updates and compute the DNN training targets from the multichannel recordings directly, instead of their single-channel version. In addition, we present comprehensive evaluation results comparing the performance of the proposed systems to other techniques.

The rest of this article is organized as follows. Section II describes the iterative EM algorithm for multichannel source separation, which is the basis for the proposed DNN-based iterative algorithm described in Section III. Section IV presents the application of the proposed framework to a music separation problem. Finally, Section V concludes the article.

II. Background

A. Problem formulation

Following classical source separation terminology [3], let I denote the number of channels, J the number of sources, $\mathbf{c}_j(t) \in \mathbb{R}^{I\times 1}$ the I-channel spatial image of source j, and $\mathbf{x}(t) \in \mathbb{R}^{I\times 1}$ the observed I-channel mixture signal. Both $\mathbf{c}_j$ and $\mathbf{x}$ are in the time domain and related by $\mathbf{x}(t) = \sum_{j=1}^{J} \mathbf{c}_j(t)$. Source separation aims to recover the source spatial images $\mathbf{c}_j(t)$ from the observed mixture signal $\mathbf{x}(t)$.

B. Model

Let $\mathbf{x}_{fn} \in \mathbb{C}^{I\times 1}$ and $\mathbf{c}_{jfn} \in \mathbb{C}^{I\times 1}$ be the short-time Fourier transform (STFT) coefficients of $\mathbf{x}$ and $\mathbf{c}_j$, respectively, for frequency bin f and time frame n. Also, let F be the number of frequency bins and N the number of frames.

[1] See MUS 2015 task on http://sisec.inria.fr.

We assume that the images $\mathbf{c}_{jfn}$ of the sources are independent of each other and follow a multivariate complex-valued zero-mean Gaussian distribution [14], [15], [17], [21]:

$$\mathbf{c}_{jfn} \sim \mathcal{N}_c\left(\mathbf{0}, v_{jfn}\mathbf{R}_{jf}\right), \tag{1}$$

$$\mathbf{x}_{fn} \sim \mathcal{N}_c\left(\mathbf{0}, \sum_{j=1}^{J} v_{jfn}\mathbf{R}_{jf}\right), \tag{2}$$

where $v_{jfn} \in \mathbb{R}_+$ denotes the power spectral density (PSD) of source j for frequency bin f and frame n, and $\mathbf{R}_{jf} \in \mathbb{C}^{I\times I}$ is the spatial covariance matrix of source j for frequency bin f. This $I \times I$ matrix represents spatial information by encoding the spatial position and the spatial width of the source [14].

Given the PSDs v_{jfn} and the spatial covariance matrices $\mathbf{R}_{jf}$ of all sources, the spatial source images can be estimated in the minimum mean square error (MMSE) sense using multichannel Wiener filtering [14], [17]:

$$\widehat{\mathbf{c}}_{jfn} = \mathbf{W}_{jfn}\mathbf{x}_{fn}, \tag{3}$$

where the Wiener filter $\mathbf{W}_{jfn}$ is given by

$$\mathbf{W}_{jfn} = v_{jfn}\mathbf{R}_{jf}\left(\sum_{j'=1}^{J} v_{j'fn}\mathbf{R}_{j'f}\right)^{-1}. \tag{4}$$

Finally, the time-domain source estimates $\widehat{\mathbf{c}}_j\left(t\right)$ are recovered from $\widehat{\mathbf{c}}_{jfn}$ by inverse STFT.

Following this formulation, source separation becomes the problem of estimating the PSD and the spatial covariance matrices of each source. This can be achieved using an EM algorithm.

C. General iterative EM framework

The iterative EM algorithm can be divided into the E-step and the M-step. The estimated PSDs v_{jfn} are initialized in the *spectrogram initialization* step. The estimated spatial covariance matrices $\mathbf{R}_{jf}$ can be initialized by $I \times I$ identity matrices. In the E-step, given the estimated parameters v_{jfn} and $\mathbf{R}_{jf}$ of each source, the source image estimates $\widehat{\mathbf{c}}_{jfn}$ are obtained by multichannel Wiener filtering (3) and the posterior second-order raw moments of the spatial source images $\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}}$ are computed as

$$\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}} = \widehat{\mathbf{c}}_{jfn}\widehat{\mathbf{c}}_{jfn}^{H} + \left(\mathbf{I} - \mathbf{W}_{jfn}\right)v_{jfn}\mathbf{R}_{jf}, \tag{5}$$

where $\mathbf{I}$ denotes the identity matrix and $\cdot^H$ is the Hermitian transposition. In the M-step, the spatial covariance matrices $\mathbf{R}_{jf}$ are updated as

$$\mathbf{R}_{jf} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{v_{jfn}}\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}}. \tag{6}$$

The source PSDs v_{jfn} are estimated without constraints as

$$z_{jfn} = \frac{1}{I}\,\mathrm{tr}\left(\mathbf{R}_{jf}^{-1}\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}}\right), \tag{7}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. Then, they are updated according to a given spectral model by fitting v_{jfn} from z_{jfn} in the *spectrogram fitting* step. The spectrogram initialization and the spectrogram fitting steps depend on how the spectral parameters are modeled, e.g. nonnegative matrix factorization (NMF) [15] or kernel additive modelling (KAM) [17]. Here, we propose the use of DNNs for this purpose.

III. DNN-based multichannel source separation

In our algorithm, DNNs are used to model the spectral parameters and predict the source spectrograms. A DNN is used for spectrogram initialization and one or more DNNs are used for spectrogram fitting. In all cases, we consider magnitude spectra (square-root PSD) as the input and output of DNNs.

Let DNN_0 be the one for initialization. It aims at simultaneously estimating all source spectra from the observed mixture. Hence, DNN_0 inputs the square-root PSD $\left\{\sqrt{z_{xfn}}\right\}_{fn}$ of the mixture, and produces estimates $\left\{\sqrt{v_{jfn}}\right\}_{jfn}$ for the square-root PSDs of all sources.

Let DNN_l be the one for fitting at iteration l. It aims at improving the source spectra estimated at the previous iteration. Thus, DNN_l inputs the current square-root PSDs $\left\{\sqrt{z_{jfn}}\right\}_{jfn}$ of all the sources jointly, and produces a refined version $\sqrt{v_{jfn}}$. The design aspects of the DNN models are described later in Section IV-C.

For updating the spatial covariance matrices, we consider a weighted form of (6):

$$\mathbf{R}_{jf} = \left(\sum_{n=1}^{N}\omega_{jfn}\right)^{-1}\sum_{n=1}^{N}\frac{\omega_{jfn}}{v_{jfn}}\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}}, \tag{8}$$

where ω_{jfn} denotes the weight of source j for frequency bin f and frame n. When $\omega_{jfn} = 1$, (8) reduces to (6). Experience shows that introducing the weights ω_{jfn} permits to improve performance. This is because they mitigate the importance of some TF points in the estimates. In this article, we use $\omega_{jfn} = v_{jfn}$, such that $\mathbf{R}_{jf} = \left(\sum_{n=1}^{N} v_{jfn}\right)^{-1}\sum_{n=1}^{N}\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}}$ as in [18], [22]. This weighting trick increases the importance of high energy TF bins whose value of $v_{jfn}^{-1}\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}}$ is closer to the true $\mathbf{R}_{jf}$ in practice. Conversely, it is worth pointing out that in [18], [22], the spatial source images are updated by $\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}} = \widehat{\mathbf{c}}_{jfn}\widehat{\mathbf{c}}_{jfn}^{H}$, hence not fully complying with the rigorous EM update (5).

The proposed DNN-based iterative algorithm is summarized in Algorithm 1.

IV. Experimental evaluation for music separation

A. Task and dataset

The task considered in the following experiments is to separate professionally-produced music recording into their constitutive stems, namely *bass*, *drums*, *vocals* and *other*. The dataset considered is a variation of the Mixing Secret Dataset (MSD100) used for SiSEC 2015. This new Demixing Secret Dataset (DSD100)[2] comprises the same 100 full-track songs as MSD100, featuring various music genres by various artists with their corresponding sources. The notable difference lies in the fact that an important sound engineering effort was undertaken so that all mixtures of DSD100 have a good sound quality and stereophonic information. The mixing was

[2]See MUS 2016 task on http://sisec.inria.fr.

Algorithm 1 DNN-based iterative algorithm

Input:

$\mathbf{x}_{fn}$ ▷ STFT of mixture: $I \times 1$

J, K, L ▷ No. of sources, spatial updates, EM iterations

$\text{DNN}_0, \text{DNN}_1, \ldots, \text{DNN}_L$ ▷ DNN spectral models

$z_{xfn} \leftarrow \text{preprocess}\ (\mathbf{x}_{fn})$

$[v_{1fn}, v_{2fn}, \ldots, v_{Jfn}] \leftarrow \left[\text{DNN}_0\left(\sqrt{z_{xfn}}\right)\right]^2$

for $j \leftarrow 1, J$ **do**

  $\mathbf{R}_{jfn} \leftarrow I \times I$ identity matrix

for $l \leftarrow 1, L$ **do**

  for $k \leftarrow 1, K$ **do**

    for $j \leftarrow 1, J$ **do**

      $\widehat{\mathbf{c}}_{jfn} \leftarrow (3)$

      $\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}} \leftarrow (5)$

      $\mathbf{R}_{jf} \leftarrow (8)$

  for $j \leftarrow 1, J$ **do**

    $z_{jfn} \leftarrow (7)$

  $[v_{1fn}, \ldots, v_{Jfn}] \leftarrow \left[\text{DNN}_l\left(\sqrt{z_{1fn}}, \ldots, \sqrt{z_{jfn}}\right)\right]^2$

for $j \leftarrow 1, J$ **do**

  $\widehat{\mathbf{c}}_{jfn} \leftarrow (3)$

Output:

$\widehat{\mathbf{c}}_{jfn}$ ▷ STFT of sources images

achieved manually for all tracks using real Digital Audio Workstations. In any case, all mixtures and sources are stereo signals sampled at 44.1 kHz. The dataset is divided evenly into development and evaluation sets.

B. General system design

The proposed DNN-based music separation system is depicted in Fig. 1. In this evaluation, we used one DNN for spectrogram initialization (DNN_0) and another DNN for spectrogram fitting (DNN_1). The system can be divided into three main successive steps:

1) Preprocessing: The STFT coefficients were extracted using a Hamming window of length 2048 and hopsize 1024. The input of DNN_0 is computed as $\sqrt{z_{xfn}} = \text{tr}\left(\mathbf{x}_{fn}\mathbf{x}_{fn}^H\right)/I$.

2) Initialization: The initial PSDs of the sources are computed from the source magnitude spectra estimated by DNN_0.

3) Multichannel filtering: The PSDs and spatial covariance matrices of the sources are estimated and updated using the proposed iterative algorithm, in which DNN_1 is used for spectrogram fitting. In order to avoid numerical instabilities due to the use of single precision, the PSDs v_{jfn} are floored to $\delta = 10^{-5}$ in the EM iteration. Also, after $\mathbf{R}_{jf}$ is updated by (8), it is regularized by $\mathbf{R}_{jf} = \mathbf{R}_{jf}I/\,\text{tr}(\mathbf{R}_{jf}) + \delta\mathbf{I}$.

C. DNN spectral models

In this subsection, we briefly describe the design aspects of DNN spectral models. See [20] for further details.

1) Architecture: The DNNs follow a multilayer perceptron (MLP) architecture. DNN_0 has an input layer size of 2050, while DNN_1 4100. DNN_0 and DNN_1 have three and two hidden layers, respectively. These settings are chosen based on the preliminary experiments. The hidden layers and output layers of both DNNs have a size of $F \times J = 4100$, i.e. the dimension of spectra times the number of sources, with rectified linear units (ReLUs) [23]. Dropout [24] with rate 0.5 is implemented for all hidden layers.

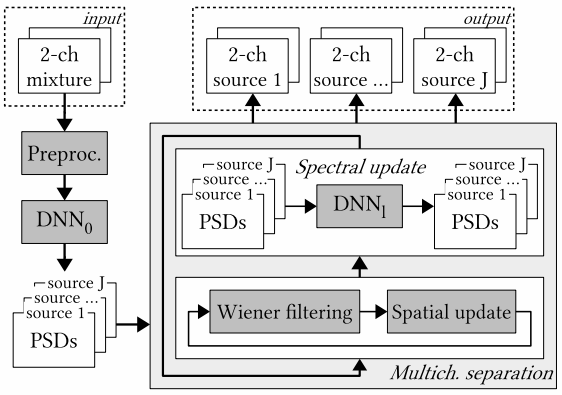

Fig. 1. Proposed DNN-based multichannel music separation framework.

2) Inputs and outputs: A *supervector* with skipped frames and complementary delta features is formed for each input frame in order to provide temporal context [20]. It consists of 2 left context, current frame, and 2 right context frames. Its dimension is then reduced by principal component analysis (PCA) to the dimension of the DNN input. Standardization is done before and after PCA. The standardization factors are computed over the development set (50 full-track songs).

3) Training criterion: The cost function used for DNN training is the sum of the mean squared error (MSE) and an ℓ_2 weight regularization term. The DNN training target is the square root of an estimate of the true spatial source image

$$\widetilde{v}_{jfn} = \frac{1}{I}\,\text{tr}\left(\widetilde{\mathbf{R}}_{jf}^{-1}\mathbf{R}_{\mathbf{c}_{jfn}}\right), \tag{9}$$

where $\widetilde{\mathbf{R}}_{jf} = \frac{1}{N}\sum_{n=1}^{N}\left(\text{tr}\left(\mathbf{R}_{\mathbf{c}_{jfn}}\right)/I\right)^{-1}\mathbf{R}_{\mathbf{c}_{jfn}}$ is an estimate of the true spatial covariance matrix and $\mathbf{R}_{\mathbf{c}_{jfn}} = \mathbf{c}_{jfn}\mathbf{c}_{jfn}^H$ is a spatial source image computed from the true source spatial image $\mathbf{c}_{jfn}$. Compared to that was done in [20], this provides better targets for the sources which are not mixed in the center (corresponding to $\widetilde{\mathbf{R}}_{jf} = \mathbf{I}$), e.g. *drums* and *other*, and consequently allows the DNN to provide better estimates.

4) Training algorithm: Most of the training aspects are the same as those in [20], including the initialization of parameters (weights and biases), the use of greedy layer-wise supervised training [25], the use of the ADADELTA parameter update algorithm [26], and the use of an early stopping mechanism. This mechanism will stop the training after 10 consecutive epochs failed to obtain better validation error and the latest model which yields the best validation error is kept. We randomly divided the supervectors of each song from the development set into training and validation sets with a ratio of 8 to 2. Although by doing so the training is prone to overfitting, it allows the training to avoid triggering the stopping mechanism too early.

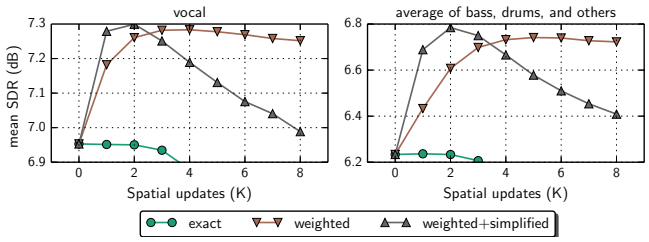

Fig. 2. Performance comparison on the *development* set for various numbers of spatial updates with different parameter updates. The PSDs v_{jfn} are estimated by the DNN_0 and the spatial covariance matrices $\mathbf{R}_{jf}$ are updated in the iterative procedure. Higher is better.

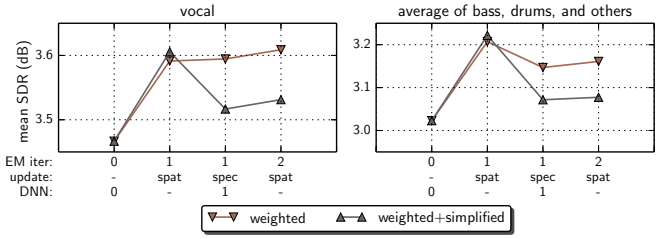

Fig. 3. Performance comparison on the *evaluation* set for each update of the EM iterations. Multiple DNNs are used. The spatial covariance matrices $\mathbf{R}_{jf}$ are updated with $K=4$ for 'weighted' and $K=2$ for 'weighted+simplified'. Higher is better.

D. Experimental results

Several ways to update the parameters are compared:

- '*exact*': $\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}} \leftarrow$ (5), $\mathbf{R}_{jf} \leftarrow$ (6) (as in [14], [15]);
- '*weighted*': $\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}} \leftarrow$ (5), $\mathbf{R}_{jf} \leftarrow$ (8) with $\omega_{jfn} = v_{jfn}$;
- '*weighted+simplified*': $\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}} = \widehat{\mathbf{c}}_{jfn}\widehat{\mathbf{c}}_{jfn}^H$; $\mathbf{R}_{jf} \leftarrow$ (8) with $\omega_{jfn} = v_{jfn}$ (as in [18], [22]).

The performance are computed on all songs on 30 seconds excerpts, taken every 15 seconds.

1) Spatial parameter updates: Fig. 2 shows the impact of the spatial parameter update strategy on the performance in terms of mean signal to distortion ratio (SDR). The number of EM iterations is fixed to $L=1$ and the spectral parameter update (lines 11-13 of Algorithm 1) is ignored. The results show that 'exact' does not work. Our preliminary experiments (not shown here) show that 'exact' does work in the oracle setting, in which v_{jfn} is computed from the true source image. Thus, in this case, 'exact' probably does not work because of bad estimation of v_{jfn} by the DNN. Conversely, 'weighted' and 'weighted+simplified' show that the weighted spatial parameter updates handle bad estimation of v_{jfn} effectively. They work well with different behavior. Using a proper number of spatial updates, 'weighted+simplified' performs better than 'weighted'. However, 'weighted' is more robust to the setting of K than the other. These results also show that the proposed multichannel separation outperforms single-channel separation (corresponding to $K=0$), even when using DNN_0 only.

2) Spectral parameter updates: Fig. 3 shows the impact of the spectral parameter updates on the performance, i.e. using DNN_1 to refine the results in the second iteration of the EM algorithm. Based on Fig. 2, the number of spatial updates is fixed to $K=4$ for 'weighted' and $K=2$ for 'weighted+simplified'. The results show that this additional spectrogram fitting step does not improve the performance. Eventually, we have an overfitting problem since spectrogram fitting provides significant improvement on the development set (not shown here). We did not have any overfitting problem in speech separation [20] even without using dropout layer for DNNs. It indicates that the problem here is probably because there are more sources with higher dimension of spectra to be estimated while the development set is small and hard to be optimally exploited for DNN training.

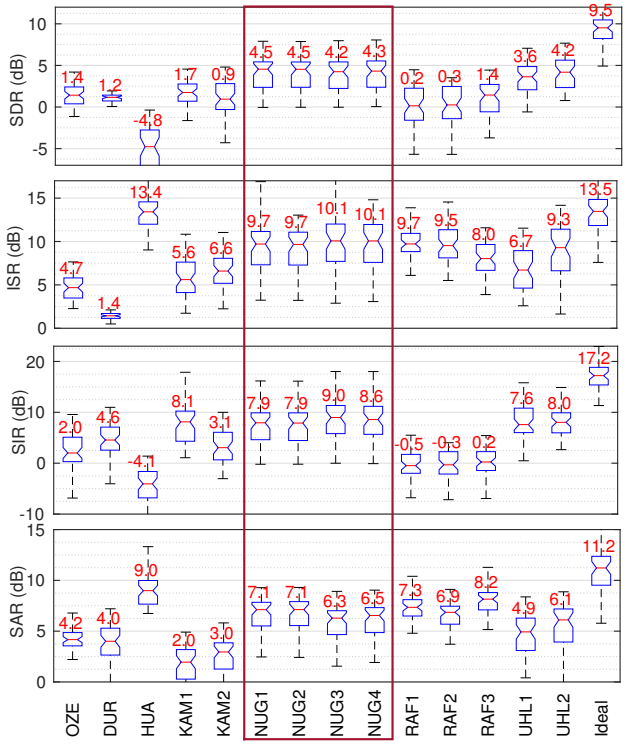

Fig. 4. Performance comparison on the *vocals* of *evaluation* set. The numbers shown above boxplots indicate the median values.

3) Comparison to other techniques: The following performance metrics are computed using BSS Eval toolbox 3.0[3] [27]: SDR, source image to spatial distortion ratio (ISR), signal to interference ratio (SIR), signal to artifacts ratio (SAR). The evaluation utilized a script provided by the challenge organizers. Fig. 4 compares the performance of the proposed system to that of other techniques on the *vocals* (see [19] for the implementation details of all techniques):

- Matrix factorization systems include `OZE` [15], `DUR` [28], and `HUA` [29];
- `KAM{1,2}` are variants of KAM [17];
- `RAF{1,2,3}` are variants of REPET [30]–[32]; and
- `UHL{1,2}` are variants of the DNN-based method in [8].

Concerning the proposed systems, `NUG1` and `NUG3` correspond to 'weighted+simplified' after spatial updates of EM iterations 1 and 2, respectively. Similarly, `NUG2` and `NUG4` correspond to 'weighted'.

The overall performance (SDR) of the proposed methods is considerably better than matrix factorization, KAM, or REPET, and slightly better than the other DNN-based method

[3] http://bass-db.gforge.inria.fr/bss_eval

[8] but not significantly so. From the performance of NUG3 and NUG4, we can observe that the use of additional DNN improves ISR and SIR, but reduces SDR and SAR. Please visit the accompanying website[4] for audio examples and code.

V. Conclusion

In this article, we presented a DNN-based multichannel source separation framework where the multichannel filter is derived using the source spectra, which are estimated by DNNs, and the spatial covariance matrices, which are updated iteratively in an EM fashion. Evaluation has been done in the context of the professionally-produced music recordings (MUS) challenge of SiSEC 2015. The proposed framework could perform well in separating singing voice and other instruments from a mixture containing multiple musical instruments. The weighted spatial parameter updates effectively handle bad estimation of spectral parameters by the DNN. The use of additional DNNs might improve the overall performance as long as overfitting can be avoided. Future works include augmenting the training data, doing formal perceptual evaluation, and estimating the optimal weights.

Acknowledgment

The authors would like to thank the developers of Theano [33]. Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr). This work is partly supported by the French National Research Agency (ANR) as parts of the DYCI2 project (ANR-14-CE24-0002-01) and the KAMoulox project (ANR-15-CE38-0003-01).

References

[1] S. Makino, H. Sawada, and T.-W. Lee, Eds., *Blind Speech Separation*. Dordrecht, The Netherlands: Springer, 2007.

[2] G. R. Naik and W. Wang, Eds., *Blind Source Separation: Advances in Theory, Algorithms and Applications*. Berlin, Germany: Springer, 2014.

[3] E. Vincent, N. Bertin, R. Gribonval, and F. Bimbot, "From blind to guided audio source separation: How models and side information can improve the separation of sound," *IEEE SPM*, vol. 31, no. 3, pp. 107–115, 2014.

[4] L. Deng and D. Yu, *Deep Learning: Methods and Applications*. Hanover, USA: Now Publishers Inc., 2014, vol. 7, no. 3-4.

[5] Y. Tu, J. Du, Y. Xu, L. Dai, and C.-H. Lee, "Speech separation based on improved deep neural networks with dual outputs of speech features for both target and interfering speakers," in *Proc. ISCSLP*, Singapore, 2014, pp. 250–254.

[6] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. ASLP*, vol. 23, no. 12, pp. 2136–2147, 2015.

[7] S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, and T. Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Proc. IEEE ICASSP*, Brisbane, Australia, 2015, pp. 116–120.

[8] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *Proc. IEEE ICASSP*, Brisbane, Australia, 2015, pp. 2135–2139.

[9] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. ASLP*, vol. 21, no. 7, pp. 1381–1390, 2013.

[10] Y. Jiang, D. Wang, R. Liu, and Z. Feng, "Binaural classification for reverberant speech segregation using deep neural networks," *IEEE/ACM Trans. ASLP*, vol. 22, no. 12, pp. 2112–2121, 2014.

[11] F. Weninger, J. Le Roux, J. R. Hershey, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. IEEE GlobalSIP*, Atlanta, USA, 2014, pp. 577–581.

[12] A. Narayanan and D. Wang, "Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training," *IEEE/ACM Trans. ASLP*, vol. 23, no. 1, pp. 92–101, 2015.

[13] Y. Wang and D. Wang, "A deep neural network for time-domain signal reconstruction," in *Proc. IEEE ICASSP*, Brisbane, Australia, 2015, pp. 4390–4394.

[14] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. ASLP*, vol. 18, no. 7, pp. 1830–1840, 2010.

[15] A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," *IEEE Trans. ASLP*, vol. 20, no. 4, pp. 1118–1133, 2012.

[16] T. Gerber, M. Dutasta, L. Girin, and C. Févotte, "Professionally-produced music separation guided by covers," in *Proc. ISMIR*, Porto, Portugal, 2012, pp. 85–90.

[17] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet, "Kernel additive models for source separation," *IEEE Trans. SP*, vol. 62, no. 16, pp. 4298–4310, 2014.

[18] A. Liutkus, D. Fitzgerald, and Z. Rafii, "Scalable audio separation with light kernel additive modelling," in *Proc. IEEE ICASSP*, Brisbane, Australia, 2015, pp. 76–80.

[19] N. Ono, D. Kitamura, Z. Rafii, N. Ito, and A. Liutkus, "The 2015 signal separation evaluation campaign (SiSEC2015)," in *Proc. LVA/ICA*, Liberec, Czech Rep., 2015.

[20] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Trans. ASLP*, to be published.

[21] E. Vincent, M. G. Jafari, S. A. Abdallah, M. D. Plumbley, and M. E. Davies, "Probabilistic modeling paradigms for audio source separation," in *Machine Audition: Principles, Algorithms and Systems*, W. Wang, Ed. Hershey, PA, USA: IGI Global, 2011, ch. 7, pp. 162–185.

[22] S. Sivasankaran, A. A. Nugraha, E. Vincent, J. A. Morales-Cordovilla, S. Dalmia, I. Illina, and A. Liutkus, "Robust ASR using neural network based speech enhancement and feature simulation," in *Proc. IEEE ASRU*, Scottsdale, USA, 2015, pp. 482–489.

[23] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proc. AISTATS*, vol. 15, Ft. Lauderdale, USA, 2011, pp. 315–323.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Research*, vol. 15, pp. 1929–1958, 2014.

[25] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. NIPS*, Vancouver, Canada, 2006, pp. 153–160.

[26] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *ArXiv e-prints*, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[27] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. ASLP*, vol. 14, no. 4, pp. 1462–1469, 2006.

[28] J.-L. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1180–1191, 2011.

[29] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. IEEE ICASSP*, Kyoto, Japan, 2012, pp. 57–60.

[30] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (REPET): A simple method for music/voice separation," *IEEE Trans. ASLP*, vol. 21, no. 1, pp. 73–84, 2013.

[31] A. Liutkus, Z. Rafii, R. Badeau, B. Pardo, and G. Richard, "Adaptive filtering for music/voice separation exploiting the repeating musical structure," in *Proc. IEEE ICASSP*, Kyoto, Japan, 2012, pp. 53–56.

[32] Z. Rafii and B. Pardo, "Music/voice separation using the similarity matrix," in *Proc. ISMIR*, Porto, Portugal, 2012, pp. 583–588.

[33] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[4] https://members.loria.fr/evincent/files/eusipco16